# UNITED STATES PATENT AND TRADEMARK OFFICE

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 09/510,203 | 02/22/2000 | Carol A. Fields | X-560 US | 2126 |

| 24309 | 7590 | 11/05/2003 |
|---|---|---|

XILINX, INC
ATTN: LEGAL DEPARTMENT
2100 LOGIC DR
SAN JOSE, CA 95124

| EXAMINER |
|---|
| SHARON, AYAL I |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2123 | |

DATE MAILED: 11/05/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) FROM
THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed
after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any
earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on <u>15 May 2003</u> .

2a)☒ This action is **FINAL**.　　2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is
closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) <u>1-19</u> is/are pending in the application.

　　4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) <u>1-9, 11, 13-14, 16-19</u> is/are rejected.

7)☒ Claim(s) <u>10,12 and 15</u> is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

　　Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

　　If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

**Priority under 35 U.S.C. §§ 119 and 120**

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

　　a)☐ All b)☐ Some * c)☐ None of:

　　　　1.☐ Certified copies of the priority documents have been received.

　　　　2.☐ Certified copies of the priority documents have been received in Application No. _____ .

　　　　3.☐ Copies of the certified copies of the priority documents have been received in this National Stage
　　　　　　application from the International Bureau (PCT Rule 17.2(a)).

　　* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

　　a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .

4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: .

## DETAILED ACTION

### *Introduction*

1. Claims 1-19 of U.S. Application 09/510,203 filed on 2/22/00 are presented for
examination. Applicants submitted a Request for Reconsideration on 8/15/03. No
changes have been made to the originally submitted claims.

### *Claim Interpretations*

2. Examiner interprets "design module" according to Applicants' definition in the
specification (p.1, lines 18-22): "System-level integration relies on reuse of
previously created designs, either from within an enterprise or from a commercial
provider. The engineering community sometimes refers to these previously
created designs as 'design modules', 'cores', or 'IP' (intellectual property)."

3. Examiner interprets "functional design element" according to Applicant's
embodiments in the specification (p.6, lines 31-33): "For example, HDL design
constructs are traced from high-level design to design elements. In schematic C,
any changes made by a translation tool are tracked." The most basic design
elements in HDL, for example, are lines, signals, and AND/OR/NOT gates.
However, these can be used to build successively higher-levels of design
elements such as flip-flops, memory arrays, adders, multiplexers, etc., all the way

up to Memory Caches and Arithmetic Logic Units (ALUs), etc. The broadest

reasonable interpretation includes all of these embodiments.

### *Allowable Subject Matter*

4. Claim 10, 12, and 15 are objected to as being dependent upon a rejected base

claim, but would be allowable if rewritten in independent form including all of the

limitations of the base claim and all intervening claims.

### *Claim Rejections - 35 USC § 102*

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that

form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

The changes made to 35 U.S.C. 102(e) by the American Inventors

Protection Act of 1999 (AIPA) and the Intellectual Property and High Technology

Technical Amendments Act of 2002 do not apply when the reference is a U.S.

patent resulting directly or indirectly from an international application filed before

November 29, 2000. Therefore, the prior art date of the reference is determined

under 35 U.S.C. 102(e) prior to the amendment by the AIPA (pre-AIPA 35 U.S.C.

102(e)).

6. The prior art used for these rejections is as follows:

7.  Fields et al., U.S. Patent 6,223,326. (Henceforth referred to as "**Fields_1**").

8.  The claim rejections are hereby summarized for Applicant's convenience. The

detailed rejections follow.

9.  **Claims 1-9, 14, and 16-19 are rejected under 35 U.S.C. 102(e) as being**

**anticipated by Fields_1.**

The applied reference has a common assignee with the instant

application.  Based upon the earlier effective U.S. filing date of the reference, it

constitutes prior art under 35 U.S.C. 102(e).  This rejection under 35 U.S.C.

102(e) might be overcome either by a showing under 37 CFR 1.132 that any

invention disclosed but not claimed in the reference was derived from the

inventor of this application and is thus not the invention "by another," or by an

appropriate showing under 37 CFR 1.131.

10. In regards to Claim 1, Fields_1 teaches the following limitations:

> 1.      A computer-implemented method for developing a reusable
> electronic circuit design module, wherein the design module
> is comprised of one or more functional design elements
> comprising the design module, comprising:
>         entering the functional design elements into a
> database;
> (Fields_1, especially: Fig.1, Item 108 and associated text)
>
>         entering documentation elements into the database;
> (Fields_1, especially: Fig.1, Item 108 and associated text)
>
>         linking the functional design elements with selected
> ones of the documentation elements;
> (Fields_1, especially: Fig.1, Item 106 and associated text)
>
>         simulating a testbench with the design module, whereby
> simulation results are generated;
> (Fields_1, especially: Fig.1, Items 110 and associated text)
>
>         storing the simulation results in the database; and
> (Fields_1, especially: Fig.1, Items 104 and associated text)

> linking the simulation results with the functional
design elements.
(Fields_1, especially: Fig.3, Items 306-318 and associated text)

11. In regards to Claim 2, Claim 1 is rejected over Fields_1 as described above. In

addition, Fields_1 teaches the following limitations:

> 2.      The method of claim 1, further comprising:
> translating the functional design elements into a
> netlist; and
> (Fields_1, especially: Fig.1, Item 102 and associated text)

> linking elements of the netlist with selected ones of
> the functional design elements.
> (Fields_1, especially: Fig.1, Item 108 and associated text)

12. In regards to Claim 3, Claim 2 is rejected over Fields_1 as described above. In

addition, Fields_1 teaches the following limitations:

> 3.      The method of claim 2, further comprising:
> translating the functional design elements into a
> physical implementation; and
> (Fields_1, especially: Fig.1, Item 102 and associated text)

> linking elements of the physical implementation with
> selected ones of the functional design elements.
> (Fields_1, especially: Fig.1, Item 108 and associated text)

13. In regards to Claim 4, Claim 1 is rejected over Fields_1 as described above. In

addition, Fields_1 teaches the following limitations:

> 4.      The method of claim 1, further comprising:
> entering simulation elements in the database; and
> (Fields_1, especially: Fig.1, Item 104 and associated text)

> linking the simulation elements to associated ones of
> the design elements.
> (Fields_1, especially: Fig.1, Items 104 and 108, and associated
> text)

14. In regards to Claim 5, Claims 1 and 4 are rejected over Fields_1 as described

above. In addition, Fields_1 teaches the following limitations:

5.      The method of claim 4, further comprising:
        entering documentation for a design script in the
database; and
(Fields_1, especially: Fig.1, Items 106 and 108, and associated text)

        linking the documentation of the design script to the
design elements comprising the design module.
(Fields_1, especially: Fig.1, Items 106 and 108, and associated text)

15. In regards to Claim 6, Claims 1 and 4 are rejected over Fields_1 as described

above. In addition, Fields_1 teaches the following limitations:

6.      The method of claim 4, further comprising:
        entering documentation for the simulation elements in
the database; and
(Fields_1, especially: Fig.1, Item 108, and associated text)

        linking the documentation for the simulation elements
with associated ones of the simulation elements.
(Fields_1, especially: Fig.1, Item 108, and associated text)

16. In regards to Claim 7, Claims 1, 4 and 6 are rejected over Fields_1 as described

above. In addition, Fields_1 teaches the following limitations:

7.      The method of claim 6, further comprising:
        inspecting the functional design elements and
simulation elements for associated documentation; and
(Fields_1, especially: Fig.1, Item 106-110, and associated text)

        reporting documentation deficiencies in association
with the functional design elements and simulation design
elements.
(Fields_1, especially: Fig.1, Item 106-110, and associated text)

17. In regards to Claim 8, Claim 1 is rejected over Fields_1 as described above. In

addition, Fields_1 teaches the following limitations:

8.      The method of claim 1, further comprising:
        inspecting the functional design elements for
associated documentation; and
(Fields_1, especially: Fig.1, Item 106-110, and associated text)

        reporting documentation deficiencies in association
with the functional design elements.
(Fields_1, especially: Fig.1, Item 106-110, and associated text)

18. In regards to Claim 9, Claim 1 is rejected over Fields_1 as described above. In

addition, Fields_1 teaches the following limitations:

> 9.      The method of claim 1, further comprising:
> inspecting the functional design elements for
> undesirable design characteristics; and
> (Fields_1, especially: Fig.1, Item 106-110, and associated text)
>
> reporting the undesirable design characteristics found
> in the functional design elements.
> (Fields_1, especially: Fig.1, Item 106-110, and associated text)

19. In regards to Claim 14, Claim 1 is rejected over Fields_1 as described above. In

addition, Fields_1 teaches the following limitations:

> 14.     The method of claim 1, further comprising:
> translating the functional design elements into a
> physical implementation; and
> (Fields_1, especially: Fig.1, Item 106-110, and associated text)
>
> linking elements of the physical implementation with
> selected ones of the functional design elements.
> (Fields_1, especially: Fig.1, Item 106-110, and associated text)

20. In regards to Claim 16, Claim 1 is rejected over Fields_1 as described above. In

addition, Fields_1 teaches the following limitations:

> 16.     The method of claim 1, further comprising displaying
> the functional design elements linked to errors in the
> simulation results.
> (Fields_1, especially: Fig.1, Item 106-110, and associated text and
> Fig.3, Items 306-318 and associated text)

21. In regards to Claim 17, Claims 1 and 16 are rejected over Fields_1 as described

above. In addition, Fields_1 teaches the following limitations:

> 17.     The method of claim 16, further comprising displaying
> documentation elements associated with errors in the
> simulation results.
> (Fields_1, especially: Fig.1, Item 106-110, and associated text and
> Fig.3, Items 306-318 and associated text)

22. In regards to Claim 18, Fields_1 teaches the following limitations:

18.    An apparatus for developing a reusable electronic circuit design module, wherein the design module is comprised of one or more functional design elements comprising the design module, comprising:
    means for entering the functional design elements into a database;
(Fields_1, especially: Fig.1, Item 108 and associated text)

    means for entering documentation elements into the database;
(Fields_1, especially: Fig.1, Item 108 and associated text)

    means for linking the functional design elements with selected ones of the documentation elements;
(Fields_1, especially: Fig.1, Item 106 and associated text)

    means for simulating a testbench with the design module, whereby simulation results are generated;
(Fields_1, especially: Fig.1, Items 110 and associated text)

    means for storing the simulation results in the. database; and
(Fields_1, especially: Fig.1, Items 104 and associated text)

    means for linking the simulation results with the functional design elements.
(Fields_1, especially: Fig.3, Items 306-318 and associated text)

23. In regards to Claim 19, Fields_1 teaches the following limitations:

19.    A system for developing a reusable electronic circuit design module, wherein the design module is comprised of one or more functional design elements comprising the design module, comprising:
    a database arranged for storage of the design elements and documentation elements;
(Fields_1, especially: Fig.1, Item 108 and associated text)

    a design inspector coupled to the database, the design inspector configured and arranged to link the functional design elements with selected ones of the documentation elements;
(Fields_1, especially: Fig.1, Item 106 and associated text)

    a debugging-support module coupled to the simulator and to the database, the debugging-support module configured and arranged to generate a netlist from the design module, wherein the netlist is suitable for simulation;
(Fields_1, especially: Fig.1, Items 110 and associated text)

    a functional simulator coupled to the debugging-support

module, the simulator configured and arranged to simulate a
testbench with the design module, whereby simulation results
are generated; and
(Fields_1, especially: Fig.1, Items 104 and associated text)

wherein the debugging-support module is further
configured and arranged to store the simulation results in
the database and link the simulation results with the
functional design elements.
(Fields_1, especially: Fig.3, Items 306-318 and associated text)

### Claim Rejections - 35 USC § 103

24. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set
forth in section 102 of this title, if the differences between the subject matter sought to be patented and
the prior art are such that the subject matter as a whole would have been obvious at the time the
invention was made to a person having ordinary skill in the art to which said subject matter pertains.
Patentability shall not be negatived by the manner in which the invention was made.

25. The prior art used for these rejections is as follows:

26. Gentry. U.S. Patent 5,673,199. (Henceforth referred to as "**Gentry**").

27. Doughty, F. "6.111 Introductory Digital Systems Laboratory". Emacs Help page.

Jan. 18, 2000. (Henceforth referred to as "**Emacs**").

28. "Intro. to Synopsys to XACT M1 Design Flow", Sept. 6, 1999. (Henceforth

referred to as "**XACT**").

29. The claim rejections are hereby summarized for Applicant's convenience. The

detailed rejections follow.

30. **Claims 1, 9, 11, and 18-19 are rejected under 35 U.S.C. 103(a) as being**

**obvious over Gentry in view of Emacs.**

31. In regards to Claim 1, Gentry teaches the following limitations:

1.      A computer-implemented method for developing a reusable
electronic circuit design module, wherein the design.module
is comprised of one or more functional design elements
comprising the design module, comprising:
        entering the functional design elements into a
database;
(Gentry, especially: Fig.2, Items 34, 36, 36', and associated text)

        simulating a testbench with the design module, whereby
simulation results are generated;
(Gentry, especially: Fig.2, Item 48, and associated text)

        storing the simulation results in the database; and
(Gentry, especially: Fig.2, Items 48, 52, 30, 32, and associated text)

        linking the simulation results with the functional
design elements.
(Gentry, especially: Fig.4, Items 80, 82, 86, and associated text)

Gentry, however, does not expressly teach the following limitations:

        entering documentation elements into the database;

        linking the functional design elements with selected
ones of the documentation elements;

On the other hand, the "Emacs" reference (pp.1-2), teaches that

documentation, in the form of comments inserted into VHDL source files, was a

standard part of the VHDL language. ("You are prompted for comments after

object definitions (i.e. signals, variables, constants, ports) and after subprogram

and process specifications if 'vhdl-prompt-for-comments' is non-nil.") The widely-

used Emacs text editor contains several functional commands for adding

comment tags to a VHDL file, as described in the "Emacs" reference.

It would have been obvious to one of ordinary skill in the art at the time the

invention was made to modify Gentry with Emacs because it was well known in

the art that inserting comments into VHDL files was a standard feature of VHDL.

32. In regards to Claim 9, Claim 1 is rejected under 35 USC 103 over Gentry in view

of Emacs, as described above. However, Gentry does not expressly teach the

following limitations:

> 9.    The method of claim 1, further comprising:
>
> inspecting the functional design elements for
> undesirable design characteristics; and
>
> reporting the undesirable design characteristics found
> in the functional design elements.

On the other hand, the "Emacs" reference (pp.2 "Source File Compilation")

teaches that "The syntax of the current buffer can be analyzed by calling a VHDL

compiler". A VHDL compiler inherently inspects design elements for undesirable

design characteristics, and reports them

It would have been obvious to one of ordinary skill in the art at the time the

invention was made to modify Gentry with "Emacs" because the use of a VHDL

compiler to test for errors is integral to the use of VHDL.

33. In regards to Claim 11, Claims 1 and 9 are rejected under 35 USC 103 over

Gentry in view of Emacs, as described above. However, Gentry does not

expressly teach the following limitations:

> 11.    The method of claim 9, further comprising:
> inspecting the functional design elements for adherence
> to predefined design rules; and
>
> reporting violations of the design rules.

On the other hand, the "Emacs" reference (pp.2 "Source File Compilation")

teaches that "The syntax of the current buffer can be analyzed by calling a VHDL

compiler". A VHDL compiler inherently inspects design elements for violations of

design rules, and reports them.

It would have been obvious to one of ordinary skill in the art at the time the

invention was made to modify Gentry with "Emacs" because the use of a VHDL

compiler to test for errors is integral to the use of VHDL.

34. In regards to Claim 18, Gentry teaches the following limitations:

> 18.    An apparatus for developing a reusable electronic
> circuit design module, wherein the design module is
> comprised of one or more functional design elements
> comprising the design module, comprising:
>         means for entering the functional design elements into
> a database;
> (Gentry, especially: Fig.2, Items 34, 36, 36', and associated text)
>
>         means for simulating a testbench with the design
> module, whereby simulation results are generated;
> (Gentry, especially: Fig.2, Item 48, and associated text)
>
>         means for storing the simulation results in the.
> database; and
> (Gentry, especially: Fig.2, Items 48, 52, 30, 32, and associated text)
>
>         means for linking the simulation results with the
> functional design elements.
> (Gentry, especially: Fig.4, Items 80, 82, 86, and associated text)

Gentry, however, does not expressly teach the following limitations:

> means for entering documentation elements into the
> database;
>
>         means for linking the functional design elements with
> selected ones of the documentation elements;

Emacs (pp.1-2), on the other hand, teaches that documentation, in the

form of comments inserted into VHDL source files, was a standard part of the

VHDL language. The widely-used Emacs text editor contains several functional

commands for adding comment tags to a VHDL file, as described in the "Emacs"

reference.

It would have been obvious to one of ordinary skill in the art at the time the

invention was made to modify Gentry with Emacs because it was well known in

the art that inserting comments into VHDL files was a standard feature of VHDL.

35. In regards to Claim 19, Gentry teaches the following limitations:

> 19.      A system for developing a reusable electronic circuit
> design module, wherein the design module is comprised of one
> or more functional design elements comprising the design
> module, comprising:
>         a database arranged for storage of the design elements
> and documentation elements;

(Gentry, especially: Fig.2, Items 34, 36, 36', and associated text)

>         a debugging-support module coupled to the simulator and
> to the database, the debugging-support module configured and
> arranged to generate a netlist from the design module,
> wherein the netlist is suitable for simulation;

(Gentry, especially: Fig.2, Item 48, and associated text)

>         a functional simulator coupled to the debugging-support
> module, the simulator configured and arranged to simulate a
> testbench with the design module, whereby simulation results
> are generated; and

(Gentry, especially: Fig.2, Items 48, 52, 30, 32, and associated text)

>         wherein the debugging-support module is further
> configured and arranged to store the simulation results in
> the database and link the simulation results with the
> functional design elements.

(Gentry, especially: Fig.4, Items 80, 82, 86, and associated text)

Gentry, however, does not expressly teach the following limitations:

> a database arranged for storage of the documentation elements;

>         a design inspector coupled to the database, the design
> inspector configured and arranged to link the functional
> design elements with selected ones of the documentation
> elements;

Emacs (pp.1-2), on the other hand, teaches that documentation, in the form of comments inserted into VHDL source files, was a standard part of the VHDL language. The widely-used Emacs text editor contains several functional commands for adding comment tags to a VHDL file, as described in the "Emacs" reference.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Gentry with Emacs because it was well known in the art that inserting comments into VHDL files was a standard feature of VHDL.

36. **Claims 2-3 and 13-14 are rejected under 35 U.S.C. 103(a) as being obvious over Gentry in view of Emacs and further in view of XACT.**

37. In regards to Claim 2, Claim 1 is rejected under 35 USC 103 over Gentry in view of Emacs, as described above. Gentry, however, does not expressly teach the following limitations:

> 2.      The method of claim 1, further comprising:
>         translating the functional design elements into a
> netlist; and
>
>         linking elements of the netlist with selected ones of
> the functional design elements.

By applicant's own admission (p.6, line 34 to p.7, line 3), translating functional design elements into a physical implementation (a netlist, for example), by using DC2NCF, NGD2VHDL, and NGD2VER is old and well known in the art.

XACT teaches that NGD2VHDL and NGD2VER are used "to create a VHD or VER file that can be simulated for back annotation within Synopsys" (XACT, p.1). The term "back annotation" is interpreted as meaning that

NGD2VHDL and NGD2VER are used to link the netlist with the functional design

elements in Synopsys.

It would have been obvious to one of ordinary skill in the art at the time the

invention was made to modify Gentry with XACT, because doing so to "to create

a VHD or VER file that can be simulated for back annotation within Synopsys"

would enable keeping both versions updated whenever changes were made in

one of the files.

38. In regards to Claim 3, Claim 2 is rejected under 35 USC 103 over Gentry in view

of Emacs, as described above. Gentry, however, does not expressly teach the

following limitations:

> 3.      The method of claim 2, further comprising:
> translating the functional design elements into a
> physical implementation; and
>
> linking elements of the physical implementation with
> selected ones of the functional design elements.

By applicant's own admission (p.6, line 34 to p.7, line 3), translating

functional design elements into a physical implementation (a netlist, for example),

by using DC2NCF, NGD2VHDL, and NGD2VER is old and well known in the art.

XACT teaches that NGD2VHDL and NGD2VER are used "to create a

VHD or VER file that can be simulated for back annotation within Synopsys"

(XACT, p.1). The term "back annotation" is interpreted as meaning that

NGD2VHDL and NGD2VER are used to link the netlist with the functional design

elements in Synopsys.

It would have been obvious to one of ordinary skill in the art at the time the

invention was made to modify Gentry with XACT, because doing so to "to create

a VHD or VER file that can be simulated for back annotation within Synopsys"

would enable keeping both versions updated whenever changes were made in

one of the files.

39. In regards to Claim 14, Claim 1 is rejected under 35 USC 103 over Gentry in

view of Emacs, as described above. However, Gentry does not expressly teach

the following limitations:

> 14.     The method of claim 1, further comprising:
>          translating the functional design elements into a
> physical implementation; and
>
>          linking elements of the physical implementation with
> selected ones of the functional design elements.

By applicant's own admission (p.6, line 34 to p.7, line 3), translating

functional design elements into a physical implementation (a netlist, for example),

by using DC2NCF, NGD2VHDL, and NGD2VER is old and well known in the art.

XACT teaches that NGD2VHDL and NGD2VER are used "to create a

VHD or VER file that can be simulated for back annotation within Synopsys"

(XACT, p.1). The term "back annotation" is interpreted as meaning that

NGD2VHDL and NGD2VER are used to link the netlist with the functional design

elements in Synopsys.

It would have been obvious to one of ordinary skill in the art at the time the

invention was made to modify Gentry with XACT, because doing so to "to create

a VHD or VER file that can be simulated for back annotation within Synopsys"

would enable keeping both versions updated whenever changes were made in
one of the files.

### Response to Arguments

### Re: 35 USC 102(e) Rejections, Fields-1 reference

40. In regards to the 35 USC 102 (e) rejection of Claim 1 by the Fields-1 reference,
Applicants unpersuasively assert (paper #4, p.3) that "... the cited portions of
Fields-1 (Fig.1, elements 104, 110 and associated text; and Fig.3, elements 306-
318) do not appear to mention a testbench, simulation, or storing simulation
results in any manner. These elements appear to call out a performance/density
analyzer, a replacement generator, and steps for converting a design to an
approximately equivalent design." Examiner respectfully disagrees with
Applicants' arguments. Examiner finds the two sets of terms to be functionally
equivalent.

- In regards to "testbench": According to the IEEE Standard Dictionary of
  Electrical and Electronics Terms (1996), a testbench is "An equipment
  specifically designed to provide a suitable work surface for testing a unit in a
  particular test setup under controlled conditions." Fields-1 teaches (col.4,
  lines 9-21) that:

  "Converter **102** is a conventional synthesis tool, for example, for taking an input design
  module, such as for an ASIC, and synthesizing the design module for a target programmable
  gate array. ... The output is an approximate synthesized design module that is provided as
  input to the performance/density analyzer **104**.

> Commercially available synthesizers and optimizers, such as those available from Simplicity, Synopsys, and Examplar also provide the capability to analyze an input design module for performance and density."

Examiner finds that the functionality of performance/density analyzer 104, as disclosed in the cited text, is functionally equivalent to that of a "testbench" as defined in the IEEE Standard Dictionary of Electrical and Electronics Terms (1996).

- In regards to "simulation": Fields-1 teaches (col.4, lines 9-21) that:

> "Converter **102** is a conventional synthesis tool, for example, for taking an input design module, such as for an ASIC, and synthesizing the design module for a target programmable gate array. ... The output is an approximate synthesized design module that is provided as input to the performance/density analyzer **104**.
> Commercially available synthesizers and optimizers, such as those available from Simplicity, Synopsys, and Examplar also provide the capability to analyze an input design module for performance and density."

The converter (Item 102) "tak[es] an input design module ... and synthesiz[es] the design module for a target programmable gate array." The resulting design module is a simulation of the target FPGA – it is not an actual FPGA. Moreover, the testing that is performed on this design module by the performance/density analyzer (Item 104) is not testing of the of an actual FPGA – it is the testing of the software-based design of the FPGA. Therefore, Fields-1 inherently teaches simulation.

- In regards to "storing simulation results": Fields-1 teaches (Fig.3, elements 306-318) a functional loop that reports problematic design elements and performs automatic approximate conversion of them. These "problematic design elements" constitute the results of an iterative simulation process. This is further described in the text (col.4, lines 39-65) as follows:

> "A database **108** includes examples of coding styles found to be inefficient based on past experience. Database **108** also includes example counters coded in, for example, Verilog or VHDL.
>
> Replacement code elements and/or recommendations are associated with the problematic design elements in database **108**. The replacements and recommendations are provided as input to replacement generator **110** along with the detected ones of the problematic design elements from design analyzer **106**. Replacement generator **110** operates in two modes. In the first mode, certain problematic design elements are automatically replaced with suitable alternatives ....
>
> After replacing the desired problematic design elements, replacement generator **110** provides the updated design module back as input to converter **102** for synthesis. Performance/Density analyzer **104** then estimates the performance and density of the new design and provides the results as output."

Examiner asserts storing the simulation results is inherent in Fields-1. The Performance/Density analyzer would not be able to "provide the results as output" without storing them in RAM or some form of media.

41. In regards to the 35 USC 102 (e) rejection of Claim 2 by the Fields-1 reference, Applicants unpersuasively assert (paper #4, p.3) that "... the rejection fails to show that Fields-1 identically shows the limitations that relate to translating the functional design elements into a netlist; and linking elements of the netlist with selected ones of the functional design elements." Examiner respectfully disagrees with Applicants' arguments.

- In regards to "translating the functional design elements into a netlist":

    Fields-1 teaches (col.4, lines 17-21) that "Commercially available synthesizers and optimizers, such as those available from Simplicity, Synopsys, and Exemplar also provide the capability to analyze an input design module for performance and density."

    Moreover, Fields-1 teaches (col.4, lines 20-27) that "It will be appreciated that performance is customarily measured along a critical path that is either

specified by a user or selected by an analysis tool. ...Density in this context in this context is measured as the number of gates per die." The terms "gates per die" and "critical path" are associated with netlists and not higher-level design elements. Examiner asserts that this is evidence that Fields-1 teaches the claimed limitation.

- In regards to "linking elements of the netlist with selected ones of the functional design elements":

    Fields-1 teaches (col.4, lines 32-37) that "Example problematic design elements include arithmetic logic cores, state machines, clock buffers, ... counters, ... decoders, ..." These items are functional design elements that one would expect in a "design module", as opposed to being a gate-level description as described in the paragraph immediately above. Therefore Examiner asserts that the "commercially available synthesizers and optimizers" that "provide the capability to analyze an input design module for performance and density", as taught by Fields-1 (col.4, lines 17-21), inherently perform the claimed linking functionality.

42. In regards to the 35 USC 102 (e) rejection of Claim 3 by the Fields-1 reference, Applicants unpersuasively assert (paper #4, p.4) that the rejection fails to show that Fields-1 teaches "linking elements of the physical implementation with selected ones of the functional design elements."

    Examiner respectfully disagrees. Examiner finds that the response to the argument made in the paragraph immediately above, in regards to the limitation

"linking elements of the netlist with selected ones of the functional design

elements" in Claim 2, applies to the current argument as well, because a "netlist"

is a "physical implementation" (as opposed to a "functional representation").

43. In regards to the 35 USC 102 (e) rejection of Claim 4 by the Fields-1 reference,

Applicants unpersuasively assert (paper #4, p.4) that the rejection fails to show

that Fields-1 teaches "entering simulation elements into the database" and

"linking the simulation elements to associated ones of the design elements."

Examiner respectfully disagrees. Fields-1 teaches (col.4, lines 42-44) that

"For example, database 108 includes examples of coding styles found to be

inefficient based on past experience." It is therefore inherent that new entries can

be added to the database immediately after a simulation discovers such an

inefficient element.

Moreover, Fields teaches that the replacement generator 110 replaces the

inefficient code discovered by design analyzer 106, based on the database 108.

(see col.4, lines 48-62). This is inherently a "linking [of] a simulation element" (i.e.

inefficient code) to "associated ones of the design elements", because Fields-1

teaches that "Example problematic design elements includes arithmetic logic

cores, state machines ...".

44. In regards to the 35 USC 102 (e) rejection of Claim 5 by the Fields-1 reference,

Applicants unpersuasively assert (paper #4, p.4) that "Fields-1 makes no

reference to any teaching that resembles a design script" and "Examiner cited

portions that ... are the same portions cited as teaching the limitations in claim 1

that relate to entering functional design elements and associated documentation

into a database."

Examiner refers the Applicants to col.4, lines 44-46, which teaches:

"Database 108 also includes example counters coded in, for example, Verilog or

VHDL." Verilog and VHDL are design scripting languages.

Moreover, Examiner wishes to point out that VLSI functional design

elements and associated documentation are usually stored in the form of Verilog

or VHDL design scripts and their embedded comments. This is why the same

teachings were used to reject both this limitation and the limitation in claim 1.

45. In regards to the 35 USC 102 (e) rejection of Claim 6 by the Fields-1 reference,

Applicants unpersuasively assert (paper #4, p.5) that "as explained above, the

rejection fails to show that Fields-1 teaches limitations related to simulation."

Examiner has addressed this argument in the response to Applicants arguments

regarding the rejection of Claim 1 on the basis of the Fields-1 reference.

46. In regards to the 35 USC 102 (e) rejections of Claims 7 and 8 by the Fields-1

reference, Applicants unpersuasively assert (paper #4, p.5) that "... the alleged

associated text mentions neither inspecting for documentation nor reporting

documentation deficiencies." Fields-1 teaches (col.4, lines 42-44) that "For

example, database 108 includes examples of coding styles found to be inefficient

based on past experience." Examiner finds that the subsequent inspaction based

on these database entries is a form of inspecting for documentation deficiencies.

47. Applicants did not submit an argument in regards to the 35 USC 102 (e)

rejections of Claim 9 by the Fields-1 reference. The rejection is maintained.

48. In regards to the 35 USC 102 (e) rejections of Claims 11-13 by the Fields-1

reference, Examiner has found Applicant's arguments to be persuasive, and has

withdrawn the rejections.

49. In regards to the 35 USC 102 (e) rejection of Claim 14 by the Fields-1 reference,

Applicants unpersuasively assert (paper #4, p.6) that "the rejection fails to show

that Fields-1 teaches the limitations that relate to translating the functional design

elements into a physical implementation; and linking elements of the physical

implementation with selected ones of the functional design elements." Examiner

finds that this is taught in the text associated with the Design Analyzer (Item 106)

and Performance/Density Analyzer (Item 104). See col.4, lines 13-29.

50. In regards to the 35 USC 102 (e) rejection of Claims 16-17 by the Fields-1

reference, Applicants repeat the arguments used regarding the rejection of Claim

1 by the Fields-1 reference. Examiner refers the Applicant's to Examiner's

response to these arguments, above.


*Re: 35 USC 102(e) Rejections, Aleksic reference*

51. In regards to the 35 USC 102 (e) rejection of Claims 1, 18, and 19 by the Aleksic

reference, Examiner has found Applicant's argument to be persuasive and has

withdrawn all rejections based on the Aleksic reference.

*Re: 35 USC 103(a) Rejections, Fields-1 and Fields-2 references*

52. In regards to the Fields-1 reference, Applicants have provided a " ... statement to

the effect that the application and reference were, at the time the invention was

made, owned by, or subject to an obligation of assignment to, the same

person(s) or organization(s)" in paper #4, p.12. See MPEP §706.02(l)(2) and

MPEP §706.02(l)(3) for the requirements. Examiner is therefore withdrawing the

rejections of Claims 10 and 15 based on Fields-1 in view of Fields-2.


*Re: 35 USC 103(a) Rejections, Gentry and Emacs references*

53. In regards to the rejections of independent claims 1, 18, and 19, Applicants

unpersuasively assert that the "there is no appearant mention of where the

simulation results are stored" in the Gentry reference. Examiner respectfully

disagrees. Applicants are referred to Fig.2, Items 36, 36', and 40. Examiner also

refers the Applicants to the "Design & Verify" Item (Fig.2, Item 42), which is the

simulation step.

54. In regards to the rejection of claim 4, Applicants assert that the "Gentry neither

shows nor suggests putting the simulation elements in the database and linking

them to the design elements." Examiner finds Applicant's argument to be

persuasive and has withdrawn the rejection of Claim 4 and its dependent claims

5-7 that were based on Gentry in view of Emacs.

55. In regards to the rejection of claim 8, Applicants argue that Gentry does not teach

"associated documentation." Examiner finds Applicant's argument to be

persuasive and has withdrawn the rejection of Claim 4 based on Gentry in view of Emacs.

56. Applicants did not submit an argument in regards to the 35 USC 103(a) rejection of Claim 9 based on Gentry in view of Emacs. The rejection is maintained.

57. In regards to the rejection of claim 10, Applicants argue that Gentry does not teach "hierarchical characteristics." Examiner finds Applicant's argument to be persuasive and has withdrawn the rejection of Claim 10 based on Gentry in view of Emacs.

58. Applicants did not submit an argument in regards to the 35 USC 103(a) rejection of Claim 11 based on Gentry in view of Emacs. The rejections is maintained.

59. In regards to the rejection of claim 12, Applicants argue that Gentry teaches "Checking VHDL syntax", but does not teach "design rules", which are more flexible. Examiner finds Applicant's argument to be persuasive and has withdrawn the rejection of Claim 12 based on Gentry in view of Emacs.

60. In regards to the rejection of claim 15, Applicants argue that Gentry does not teach "the hierarchy of the design module." Examiner finds Applicant's argument to be persuasive and has withdrawn the rejection of Claim 15 based on Gentry in view of Emacs.

61. In regards to the rejection of claims 16-17, Examiner finds Applicant's argument to be persuasive and has withdrawn the rejection of Claims 16-17 based on Gentry in view of Emacs.

*Re: 35 USC 103(a) Rejections, Gentry, Emacs and XACT references*

62. In regards to the rejections of independent claims 2-3 and 14, Applicants

unpersuasively assert (paper #4, p.16) that "The rejection is respectfully

traversed because *prima facie* obviousness is not established." More specifically,

applicants assert that "Claims 2 and 14 depend from claim 1, and claim 3

depends from claim 2. As explained above, the rejection fails to establish a *prima*

*facie* case of obviousness of claim 1 in view of the Gentry-Emacs combination.

Therefore, for at least these reasons *prima facie* obviousness is not established

for claims 2 and 3."

Examiner responded to this argument in paragraph 53 of this office action,

and has maintained the rejection of claim 1 in view of the Gentry-Emacs

combination.

63. In regards to the rejection of claim 13, Examiner finds Applicant's argument to be

persuasive and has withdrawn the rejection of Claim 13 based on Gentry in view

of Emacs and further in view of XACT.

### Conclusion

64. Applicant's arguments filed 5/15/03 have been fully considered but they are not

persuasive.

65. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time

policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

### *Correspondence Information*

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ayal I. Sharon whose telephone number is (703) 306-0297. The examiner can normally be reached on Monday through Thursday, and the first Friday of a biweek, 8:30 am – 5:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kevin Teska can be reached on (703) 305-9704. Any response to this office action should be mailed to:

Director of Patents and Trademarks
Washington, DC 20231

Hand-delivered responses should be brought to the following office:

4<sup>th</sup> floor receptionist's office
Crystal Park 2
2121 Crystal Drive
Arlington, VA

The fax phone numbers for the organization where this application or proceeding
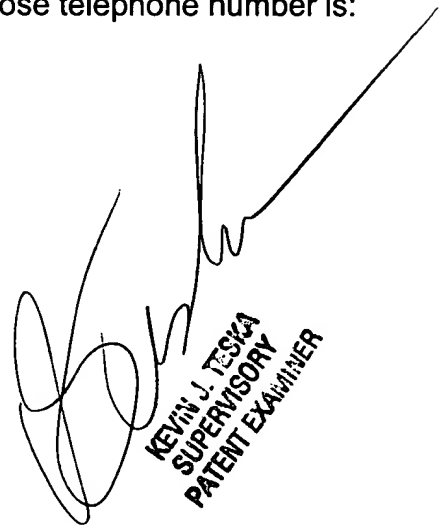
is assigned are:

All communications:        (703) 872-9306

Any inquiry of a general nature or relating to the status of this application

or proceeding should be directed to the receptionist, whose telephone number is:

(703) 305-3900.


Ayal I. Sharon

Art Unit 2123

October 20, 2003